

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**DETERMINING REGIONS THAT ARE OCCLUDED
FROM AN OBSERVATION POINT**

Inventor(s):

Andrew M. Vincent

ATTORNEY'S DOCKET NO. MS1-1029US

10014879-134101
TOP SECRET 628400T

1 **CROSS-REFERENCE TO RELATED APPLICATIONS**

2
3 The present patent application claims the benefit of U.S. Provisional
4 Application No. 60/255,883, filed December 18, 2000.
5

6 **TECHNICAL FIELD**

7 This application relates to computer graphics and image processing. More
8 particularly, the present application relates to a method and system for determining
9 portions of a geographic terrain surface that are occluded from an observation
10 point.
11

12 **BACKGROUND**

13 Geographic terrain defines the dynamics of operations involving the
14 monitoring of objects through sensors, such as radar. For example, a terrain's
15 shape can create regions that are occluded (i.e., not visible) from a sensor placed at
16 a particular location. These occluded regions inhibit the detection of objects, such
17 as vehicles and people. Furthermore, the existence of such regions undermines the
18 confidence placed by tactical operators, such as air traffic controllers, in the
19 efficacy of monitoring systems employing such sensors.

20 To effectively detect the presence and activity of objects, a working
21 knowledge of a geographic terrain is an important asset. This knowledge enables
22 an optimal placement of sensors. To aid in the development of such knowledge,
23 high-resolution digital elevation models providing precise graphical models of
24 geographic terrain are becoming increasingly available. These models enable
25 quick and efficient analysis of terrain by processing platforms, such as computers.

1 Accordingly, what is needed is a computer-based tool that, from a graphical
2 terrain model, assesses visibility. In one application, visibility needs to be
3 assessed from observation point(s) to assist operators of systems involving sensors
4 to provide effective monitoring of objects.

5 6 **SUMMARY**

7 Described herein is a system, method, and computer program product for
8 determining portions of a surface within a space that are occluded from an
9 observation point. In the described implementation, a horizon is defined with
10 reference to the observation point; and a region within the space that is beneath the
11 horizon is identified. A ray is projected from the observation point that intersects
12 the horizon and defines an area that is occluded from the observation point.

13 The described implementations, therefore, introduce the broad concept of
14 generating knowledge of terrain, which advantageously provides for the effective
15 monitoring of objects.

16 17 **BRIEF DESCRIPTION OF THE DRAWINGS**

18 In the drawings, the left-most digit(s) of a reference number identifies the
19 drawing in which the reference number first appears.

20 FIG. 1 is a cross-sectional view of a three-dimensional strategic space.

21 FIG. 2 is a flowchart illustrating an operational sequence directed to
22 identifying occluded regions.

23 FIG. 3 is a flowchart illustrating an operational sequence directed to
24 determining a horizon elevation angle.

25 FIG. 4 is a diagram illustrating a plurality of elevation angles.

FIG. 5 is a flowchart illustrating an operational sequence directed to updating an azimuth angle.

FIG. 6 is a diagram illustrating a plurality of azimuth angles.

FIG. 7 is a flowchart illustrating an operational sequence directed to the identification of occluded region(s).

FIG. 8 is a diagram of occluded regions identified at a particular azimuth angle in relation to an observation point.

FIG. 9 is a flowchart illustrating the identification of occluded regions that are beneath a horizon coordinate.

FIG. 10 is a diagram illustrating a technique of measuring occluded region depth.

FIG. 11 is a block diagram of an exemplary computer system.

DETAILED DESCRIPTION

FIG. 1 is a cross-sectional view of a three-dimensional strategic space 100. Strategic space 100 includes a terrain surface 102 and an exemplary observation point 104. Terrain surface 102 is a three-dimensional surface defined using a triangle mesh. However, other surface models, such a quadrilateral mesh, can be used. Terrain surface 102 includes a plurality of peaks (also referred to herein as apexes) 106a-c.

As shown in FIG. 1, observation point 104 is not disposed on terrain surface 102. Rather, observation point 104 is above terrain surface 102. Thus, observation point 104 is supported by an aerial platform, such as an airplane, a helicopter, a missile, or a satellite. Although observation point 104 is above

1 terrain surface 102, other observation point(s) can be disposed on terrain surface
2 102, and beneath terrain surface 102 (i.e., in subterranean locations).

3 Three exemplary lines of sight (LOSs) 108, 110, and 118 are projected
4 from observation point 104. Each of LOSs 108, 110, and 118 indicates portions of
5 terrain surface 102 that obstruct the collection of tactical data by an image
6 gathering sensor located at observation point 104. Examples of image gathering
7 sensors include visible light sensors, infrared (IR) light sensors, ultraviolet (UV)
8 light sensors, and radar.

9 LOSs 108, 110, and 118 show that terrain surface 102 occludes regions
10 from observation point 104. For instance, FIG. 1 illustrates three such occluded
11 regions 112, 114, and 116.

12 Occluded regions 112, 114, and 116, as illustrated in FIG. 1, are cross-
13 sectional areas that represent portions of three-dimensional strategic space 100.
14 These portions are not visible to an observer (i.e., a sensor) located at observation
15 point 104.

16 The described exemplary implementations herein provide a technique for
17 identifying occluded cross-sectional areas. From such cross-sectional areas, three-
18 dimensional volumes that are occluded from an observation point, such as
19 observation point 104, can be derived.

20 FIG. 2 is a flowchart illustrating an operational sequence directed to
21 identifying occluded regions (steps 202-216). The steps in FIG. 2 are described
22 with reference to terrain surface 102 and observation point 104. However, these
23 steps can be applied to other terrains, surfaces, and observation points, as would be
24 apparent to persons skilled in the relevant art(s).

The operation illustrated with reference to FIG. 2 commences with a step 202. In this step, a user selects an observation point, such as observation point 104. As described above with reference to FIG. 1, this observation point can be disposed on, above, or beneath terrain surface 104.

Next, in a step 204, a horizon contour on terrain surface 102 is defined with reference to observation point 104. This step comprises generating a plurality of horizon coordinates, where each horizon coordinate includes an azimuth angle and a elevation angle. Together, these horizon coordinates provide a set of samples that form a horizon contour. This horizon contour azimuthally surrounds observation point 104. An exemplary horizon contour is described below with reference to FIG. 6.

By using observation point 104 as the origin of a spherical coordinate system, an azimuth angle, an elevation angle, and a radius define a point in three-dimensional strategic space 100. These values can be used to form a coordinate system to define strategic space 100.

As shown in FIG. 2, performance of step 204 comprises the performance of steps 206 through 214. In step 206, an initial azimuth angle is selected. An azimuth angle is a horizontal angle measured from a reference direction, usually due north, of a ray that is projected from observation point 104.

Next, in a step 208, a horizon elevation angle and a horizon radius are determined from the observation point selected in step 202. An elevation angle is an angle that a ray projected from observation point 104 makes with a horizontal plane. For a particular azimuth angle, a horizon elevation angle is the largest elevation angle at which a ray projected from observation point 104 intersects an apex on surface 102. For such an intersection, the horizon radius is the distance

1 along the projected ray from observation point 104 to the point of intersection.
 2 Performance of step 208 is described in greater detail below with reference to FIG.
 3 3.

4 In a step 210, the azimuth angle, horizon elevation angle, and horizon
 5 radius are stored as one of a plurality of horizon coordinates. Together, this
 6 plurality of horizon coordinates defines a horizon contour.

7 A step 212 follows the performance of step 210. In this step, it is
 8 determined whether to continue with the generation of horizon coordinates. If the
 9 generation of horizon coordinates is to continue, then a step 214 is performed.
 10 Otherwise, operation proceeds to a step 216. For example, horizon coordinates
 11 can continue to be generated until a complete (or nearly complete) horizon contour
 12 has been formed. In this case, step 212 comprises the step of determining whether
 13 a complete horizon contour has been formed. A complete horizon contour has
 14 been formed if the azimuth angle of the most recently stored horizon coordinate is
 15 within a predetermined threshold of a 360 degrees azimuthal rotation from the
 16 initial azimuth angle selected in step 206.

17 In a step 214, the azimuth angle stored during the most recent performance
 18 of step 210 is updated. Control then proceeds to continue the generation of a new
 19 horizon coordinate through the performance of steps 206 through 210. The
 20 performance of step 214 is described in greater detail with reference to FIG. 5.
 21 After completion of step 214, operation returns to step 208.

22 As described above, if the generation of horizon coordinates pursuant to
 23 step 204 is complete, then a step 216 is performed. In step 216, occluded
 24 region(s), such as regions 112, 114, and 116, are identified.
 25

FIG. 3 is a flowchart illustrating the performance of step 208 in greater detail. As described above, for a particular azimuth angle, a horizon elevation angle is determined in this step. This operation involves an iterative process that calculates high, low, and middle elevation angles. These angles are repeatedly calculated until the middle elevation angle converges upon a horizon elevation angle. This iterative approach is referred to herein as binary adaptive searching.

Performance of step 208 begins with a step 302. In this step, an iteration index is set to an initial value, such as zero.

Next, in a step 304, high and low elevation angles $e_{high}(i)$ and $e_{low}(i)$ are initialized. This step comprises setting $e_{high}(i)$ to an initial value of +90 degrees and $e_{low}(i)$ to an initial value of -90 degrees. However, other initial values may be employed.

In step 306, a corresponding middle elevation angle, $e_{mid}(i)$, is calculated. This calculation is made according to Equation (1).

$$e_{mid}(i) = \frac{e_{low}(i) + e_{high}(i)}{2} \quad (1)$$

In a step 308, a ray is projected from observation point 104 at the current azimuth angle and elevation angle $e_{mid}(i)$.

In a step 310, it is determined whether the ray projected in step 308 intersects terrain surface 102. If such an intersection occurred, then a step 312 is performed. Otherwise, a step 314 is performed.

In step 312, the low elevation angle, $e_{low}(i)$, is adjusted. Performance of this step comprises setting the value of $e_{low}(i)$ to the value of $e_{mid}(i)$.

1 In step 314, $e_{high}(i)$ is adjusted. This adjustment includes setting the value
2 of $e_{high}(i)$ to the value of $e_{mid}(i)$.

3 After the performance of either step 312 or step 314, a step 316 is
4 performed. Step 316 determines whether $e_{mid}(i)$ has converged upon a horizon
5 elevation angle. Performance of step 316 comprises the step of determining
6 whether the difference between the current middle elevation angle, $e_{mid}(i)$, and the
7 previous middle elevation angle, $e_{mid}(i-1)$, is less than a convergence threshold, C .
8 This determination is expressed below in Equation (2).

$$|e_{mid}(i) - e_{mid}(i-1)| < C \quad (2)$$

9
10
11
12 During a first performance of step 316 for a particular azimuth angle, step
13 316 will not indicate convergence.

14 If $e_{mid}(i)$ has not converged upon a horizon elevation angle, then a step 318
15 is performed. In step 318, iteration index, i , is incremented by one. After i is
16 incremented, steps 306-316 are repeated.

17 If $e_{mid}(i)$ has converged upon a horizon elevation angle, then a step 320 is
18 performed. In step 320, the horizon elevation angle that corresponds to the current
19 azimuth angle is set. In addition, the horizon radius is set to the intersection radius
20 of the ray projected in step 308.

21 FIG. 4 is a diagram that illustrates the iterative process that is described
22 above with reference to FIG. 3. In particular this diagram shows a plurality of
23 rays 402-416 projected from observation point 104. Each of these rays has the
24 same azimuth angle. However, these rays are projected at elevation angles that
25 equal $e_{low}(i)$, $e_{mid}(i)$, and $e_{high}(i)$ during various iterations within step 208.

Table 1 is provided below to illustrate an exemplary iterative performance of step 208. In this table, each row represents an iteration, and includes three columns that indicate which of rays 402-416 correspond to $e_{low}(i)$, $e_{mid}(i)$, and $e_{high}(i)$ for the represented iteration.

Iteration Index, i	$e_{low}(i)$	$e_{mid}(i)$	$e_{high}(i)$
1	416	414	402
2	414	404	402
3	414	406	404
4	414	412	406
5	412	410	406
6	410	408	406

Table 1

As shown in Table 1, $e_{mid}(i)$ converges at the projection of ray 408. This convergence occurred during six iterations of the process described above with reference to FIG. 3.

As described above with reference to step 204 of FIG. 2, a plurality of horizon coordinates are generated to form a horizon contour. Each of these horizon coordinates includes an azimuth angle and an elevation angle. In step 214, which is within step 204, an azimuth angle is updated for generation of a corresponding horizon elevation angle and a corresponding horizon radius. These values are then stored as a horizon coordinate.

FIG. 5 is a flowchart illustrating an operation of step 214 in greater detail. This operation advantageously enables the spacing of horizon coordinates across a horizon contour in a substantially uniform manner. This operation begins with a step 502. In step 502 the horizon radius corresponding to the current azimuth angle, $r_{horizon}$ is obtained. Next, in a step 504, a desired separation distance, $d_{desired}$, is obtained. This desired separation distance can be predetermined or user-selectable, and indicates a desired separation expressed as a Euclidean distance in units such as kilometers.

A step 506 follows the performance of step 504. In step 506, an azimuth angle increment, Δaz , is calculated. This calculation is based on the horizon radius obtained in step 502 and the desired separation distance obtained in step 504. Performance of step 506 is expressed below in Equation (3).

$$\Delta az = d_{desired} * r_{horizon} \quad (3)$$

1 Next, in a step 508, the azimuth angle increment, Δaz , is added to the
2 current azimuth angle to obtain an updated azimuth angle.

3 FIG. 6 is a diagram illustrating an overhead view of an exemplary horizon
4 contour 600 from the perspective of observation point 104. As described herein,
5 horizon contour 600 includes a plurality of horizon coordinates. These
6 coordinates are shown in FIG. 6 as coordinates 602-634. As described herein,
7 each of these coordinates has an azimuth angle, a horizon elevation angle, and a
8 horizon radius.

9 Horizon coordinates 610 and 612 provide an illustration of an azimuth
10 angle increment 650, (shown in FIG. 6 as $\Delta \alpha$ 650) that exists between these
11 coordinates. As shown in FIG. 6, coordinates 610 and 612 are separated by
12 maximum error distance 660. In contrast, coordinates 630 and 632 are separated
13 by a minimum error distance 662. Distances 660 and 662 may be employed to
14 ensure that spacing between adjacent horizon coordinates is within an appropriate
15 range. This distance may be a Euclidean distance, or any other type of distance, as
16 would be apparent to persons skilled in the relevant art(s). If the distance between
17 two adjacent horizon coordinates is greater than maximum error distance 660 or
18 less than minimum error distance 662, then the one or both of these coordinates
19 can be recalculated.

20 FIG. 7 is a flowchart illustrating the performance of step 216 in greater
21 detail (steps 702-714). This operational sequence begins with a step 702. In step
22 702 one or more occluded cross-sectional areas are identified. Step 702 includes
23 steps 704 through 710. To aid in visualization, step 702 is described with
24 reference to FIG. 8, which is a diagram illustrating an exemplary side view of
25 terrain surface 102.

As described above, step 702 includes steps 704 through 710. In step 704, a horizon coordinate is selected from one of the plurality of horizon coordinates generated in step 204.

Next, in a step 706, an implicit occluded region is identified beyond the selected horizon coordinate. Such an occluded region is bounded by a horizon ray projected from observation point 104 through the selected horizon coordinate, such as ray 810 in FIG. 8. In particular, an implicit occluded region is the cross-sectional area that exists between terrain surface 102 and a portion of a horizon ray that is beyond its corresponding horizon coordinate. Referring to FIG. 8, occluded region 802 is an example of an implicit occluded region that is beyond horizon coordinate 830.

A step 708 follows step 706. In step 708, occluded regions that are beneath the selected horizon coordinate are identified. Performance of this step is an iterative process that is described in greater detail with reference to FIG. 9.

As shown in FIG. 9, step 708 commences with a step 902 of initializing an iteration index, j . Next, in a step 904, the horizon radius and the horizon elevation angle of the coordinate selected in step 704 are set to previous iteration values r_{j-1} and e_{j-1} , respectively.

A step 906 follows step 904. In this step, an elevation angle for the current iteration, e_j , is calculated. This step comprises decreasing e_{j-1} by a predetermined amount, $\Delta\theta_j$.

Next, in a step 908, a ray is projected from observation point 104 at the azimuth angle of the horizon coordinate selected in step 704 and elevation angle, e_j . In step 910, an intersection radius, r_j , is determined. This radius is the distance

1 from observation point 104 at which the ray projected in step 908 intersects terrain
2 surface 102.

3 In a step 912, r_j and r_{j-1} are compared. If the difference between these
4 values is greater than a predetermined threshold, then a step 916 is performed.
5 Otherwise, operation proceeds to a step 914. In step 914, j is incremented. After
6 step 914, operation returns to step 906.

7 In step 916, an occluded region beneath the horizon coordinate selected in
8 step 704 is defined. This occluded region is bounded by an occlusion ray
9 projected from observation point 104 at the azimuth angle of the coordinate
10 selected in step 704 and the previous elevation angle e_{j-1} , such as ray 818 in FIG.
11 8. Other suitable elevation angles may be employed. In particular, this occluded
12 region is the cross sectional area that exists between terrain surface 102 and a
13 portion of a occlusion ray that is between observation point 104 and the point
14 where it intersects terrain surface 102.

15 Referring to FIG. 8, occluded region 804 is an example of an occluded
16 region that is beneath horizon coordinate 830. Rays 810 through 820 illustrate an
17 exemplary performance of step 708. Each of rays 810 through 820 have a
18 corresponding intersection radius that indicates the distance along the ray between
19 observation point 104 and the nearest intersection with terrain surface 102. For
20 example, FIG. 8 illustrates an R_0 that corresponds to ray 810, an R_1 that
21 corresponds to ray 812, an R_2 that corresponds to ray 814, an R_3 that corresponds
22 to ray 816, an R_4 that corresponds to ray 818, and an R_5 that corresponds to ray
23 820.

24 Referring once again to FIG. 7, a step 710 is performed after step 708. In
25 step 710, it is determined whether all of the stored horizon coordinates have been

1 selected. If so, then a step 712 is performed. Otherwise, operation returns to step
2 704.

3 In step 712, any shallow occluded regions that were identified in step 702
4 are culled. This step includes disregarding the shallow region's status as an
5 occluded region. Performance of step 712 is optional. This step is described
6 below in greater detail with reference to FIG. 10.

7 After step 712, an optional step 714 is performed. In step 714, one or more
8 occluded volumes are generated. Performance of this step includes interpolating
9 volumes from the occluded cross-sectional areas identified in step 702.

10 For many applications, shallow occluded regions are of little significance.
11 For example, in the context of tactical mission planning, shallow occluded regions
12 of less than two meters do not occlude the presence of relevant objects, such as
13 ground-based and aerial vehicles. Therefore, as described above with reference to
14 step 712, this exemplary implementation provides for shallow occluded regions to
15 be culled.

16 FIG. 10 is a diagram illustrating a technique of measuring a depth of an
17 occluded region 1000. In FIG. 10, an occlusion ray 1002 generated as described
18 above with reference to FIGs. 7 and 9 defines an edge of occluded region 1000.
19 The depth of occluded region 1000 is determined by measuring the distance
20 between occlusion ray 1002 and terrain surface 102 at various locations within
21 occluded region 1000.

22 These locations are points spaced along occlusion ray 1002. As shown in
23 FIG. 10, points 1004, 1005, 1006, 1008, 1010, 1012, and 1014 are uniformly
24 spaced by separation distance 1020. However, other spacing schemes, may be
25 employed. At each of these points, a tangential distance is measured from

1 occlusion ray 1002 to terrain surface 102. If none of these distances are greater
2 than minimum depth 1022, then the occluded region is identified as a shallow
3 occluded region and culled.

4 The concepts described herein may be implemented using hardware,
5 software or a combination thereof and may be implemented in a computer system
6 or other processing system. An exemplary computer system 1101 implementation
7 is shown in FIG. 11. Computer system 1101 includes one or more processors,
8 such as a processor 1104. The processor 1104 is connected to a communication
9 bus 1102. Various software embodiments are described in terms of this example
10 computer system. After reading this description, it will become apparent to
11 persons skilled in the relevant art how to implement the invention using other
12 computer systems and/or computer architectures.

13 Computer system 1102 also includes a main memory 1106, preferably
14 random access memory (RAM), and can also include a secondary memory 1108.
15 The secondary memory 1108 can include, for example, a hard disk drive 1110
16 and/or a removable storage drive 1112, representing a floppy disk drive, a
17 magnetic tape drive, an optical disk drive, etc. The removable storage drive 1112
18 reads from and/or writes to a removable storage unit 1114 in a well known
19 manner. Removable storage unit 1114, represents a floppy disk, magnetic tape,
20 optical disk, etc. which is read by and written to by removable storage drive 1112.
21 As will be appreciated, the removable storage unit 1114 includes a computer
22 usable storage medium having stored therein computer software and/or data.

23 In alternative embodiments, secondary memory 1108 may include other
24 similar means for allowing computer programs or other instructions to be loaded
25 into computer system 1101. Such means can include, for example, a removable

1 storage unit 1122 and an interface 1120. Examples of such can include a program
2 cartridge and cartridge interface (such as that found in video game devices), a
3 removable memory chip (such as an EPROM, or PROM) and associated socket,
4 and other removable storage units 1122 and interfaces 1120 which allow software
5 and data to be transferred from the removable storage unit 1122 to computer
6 system 1101.

7 Computer system 1101 can also include a communications interface 1124.
8 Communications interface 1124 allows software and data to be transferred
9 between computer system 1101 and external devices. Examples of
10 communications interface 1124 can include a modem, a network interface (such as
11 an Ethernet card), a communications port, a PCMCIA slot and card, etc. Software
12 and data transferred via communications interface 1124 are in the form of signals
13 which can be electronic, electromagnetic, optical or other signals capable of being
14 received by communications interface 1124. These signals 1126 are provided to
15 communications interface via a channel 1128. This channel 1128 carries signals
16 1126 and can be implemented using wire or cable, fiber optics, a phone line, a
17 cellular phone link, an RF link and other communications channels.

18 In this document, the terms "computer program medium" and "computer
19 usable medium" are used to generally refer to media such as removable storage
20 device 1112, a hard disk installed in hard disk drive 1110, and signals 1126.
21 These computer program products are means for providing software to computer
22 system 1101.

23 Computer programs (also called computer control logic) are stored in main
24 memory and/or secondary memory 1108. Computer programs can also be
25 received via communications interface 1124. Such computer programs, when

1 executed, enable the computer system 1101 to perform the features of the present
2 invention as discussed herein. In particular, the computer programs, when
3 executed, enable the processor 1104 to perform the features of the present
4 invention. Accordingly, such computer programs represent controllers of the
5 computer system 1101.

6 In an embodiment where the invention is implemented using software, the
7 software may be stored in a computer program product and loaded into computer
8 system 1101 using removable storage drive 1112, hard drive 1110 or
9 communications interface 1124. The control logic (software), when executed by
10 the processor 1104, causes the processor 1104 to perform the functions described
11 herein.

12 In another exemplary description, hardware components such as application
13 specific integrated circuits (ASICs) can be used to implement the functionality
14 described herein. Implementation of the hardware state machine so as to perform
15 the functions described herein will be apparent to persons skilled in the relevant
16 art(s).

17 In yet another embodiment, the invention is implemented using a
18 combination of both hardware and software. Examples of such combinations
19 include, but are not limited to, microcontrollers.

20 **Conclusion**

21 Although the invention has been described in language specific to structural
22 features and/or methodological acts, it is to be understood that the invention
23 defined in the appended claims is not necessarily limited to the specific features or
24 acts described. Rather, the specific features and acts are disclosed as exemplary
25 forms of implementing the claimed invention.